Application of Learning Pallets for Real-Time Scheduling by Use of Artificial Neural Network

A. Mehrsai[1], H.R. Karimi[2], I. Rügge [1], B. Scholz-Reiter[1]

[1] Department of Production Engineering, Bremen University, BIBA GmbH, BIBA, Hochschulring 20, 28359 Bremen, Germany
[2]Department of Engineering, Faculty of Engineering and Science, University of Agder, Grimstad, Norway

Generally, this paper deals with the problem of autonomy in logistics. Specifically here, a complex problem in inbound logistics is considered as real-time scheduling in a stochastic shop floor problem. Recently, in order to comply with real-time decisions, autonomous logistic objects have been suggested as an alternative. Since pallets are common used objects in carrying materials (finished or semi-finished), so they have the possibility to undertake the responsibility of real time dispatching jobs to machines in a shop-floor problem. By insisting on the role of pallets for this task, their sustainment's advantage in manufacturing systems motivated the idea of developing learning pallets. These pallets may deal with uncertainties and sudden changes in the assembly system. Here, among some intelligent techniques artificial neural network is selected to transmit the ability of decision making as well as learning to the pallets, as distributed objects. Besides, pallets make decisions based on their own experiences about the entire system and local situations. Consequently, the considered scheduling problem resembles an open shop problem with three alternative finished products. Finally, a discrete event simulation model is developed to solve this problem and defined the results of this transmission paradigm.

***Index Terms*— Assembly Systems, Learning, Neural Networks, Real Time Systems.**

I. Introduction

IN GENERAL, logistics can be explained as the science of organizing and handling material flows with a wide range of operations and processes that has a crucial role in sustaining industries. Accordingly, the vast scope of activities in logistics makes it one of the most cost drivers and complex missions in production businesses. Although all activities in logistics are pertinent to and correlated for generic goals, but for the sake of simplicity logistics tasks are generally split up into inbound and outbound operations. Commonly, the outbound logistics consist of those operations and planning that organize the flow of materials between members of logistics networks, from point of origin to the point of consumption. On the other hand, inbound logistics covers every kind of activity and scheduling in production logistics which has direct or indirect effect on material flow and handling inside factories. Meanwhile scheduling and control of production operations can be considered as the core of the inbound logistics problem.

Universally, scheduling processes concentrate on optimizing material routing as well as allocation of jobs to some resources, so that all existing constrains are satisfied [1].This mission is conventionally done by identifying some limitations and assumptions in advance which includes processing times, release times of jobs, due dates, number of orders, and etc. In other words, the number of jobs and their characteristics as well as shop-floor circumstance in a scheduling problem are given or assumed to be known in the problem. In doing so, the scheduling solutions are derived in offline manners, thus, mostly can be considered as rough solutions or the idealistic targets for scheduling real problems. However, it can be seen in practice that several changes happen during a running system which are not perceivable (predictable) or difficult to consider them beforehand, e.g., breakdowns, urgent jobs, delayed supply. In fact, these changes and disturbances, called dynamics, are the causes of increasing the intricacy of practice oriented problems.

Nevertheless, several arguments are reported concerning the inability of conventional scheduling methods, by offline approach, to feasibly solving problems in practice. Alternatively, online scheduling, dynamic scheduling, and real-time scheduling problems are introduced to defeat the impediments in the way of defining a feasible solution [1] [2].For example, online scheduling makes its decisions when the system is running, without any precise information about the prospective inputs [3]. However, online scheduling does adapt its solution to the current situation within an interval rather than deciding in real-time states.

Similarly, in real-time scheduling the jobs come to an assembly system in various instances and they are supposed to be allocated to machines in real-times, usually by means of available dispatching rules. Thus, real time scheduling requires employing the advantages of online scheduling while it makes decisions in real times. In this manner, in case of technical and methodical availability, the real time scheduling problem is the prominent choice that covers the characteristics of the other choices. At the same time, it is the most suitable solution in the presence of nondeterministic events (dynamics), happening in a very short period of time with on time decision making request [4].

In addition, when a problem is enough complex, like scheduling, then simultaneously monitoring of each operation with every detail, by means of a central master, seems a very sophisticated duty. On the other hand, recently, a promising alternative is being developed that can deal with material flow and scheduling problems in a decentralized and distributed manner. The approach proposes application of self-organizing and autonomous objects to face such problems by themselves in real-time, instead of following offline schedules. This decentralized tactic is being enthusiastically suggested and, to some extent, its performance examined [4]. Competently, the real-time scheduling procedure can properly adopt the notion

of distributed autonomous objects, called holons or agents in different context, and enhance its performance [5] [6] [7].

Furthermore, inspired by pull manufacturing systems, in particular Conwip system, the carts of conveying products in an assembly system (pallets here) are retained and circulated in a closed loop system. This specification raises the concept of using pallets as distributed objects to make real-time decisions for allocation and dispatching, concerning their vicinity to the single products in manufacturing logistics. Appropriately, these autonomous pallets even can comply with the notion of customization in manufacturing, as a growing appealed contribution in manufacturing logistics [8].

However, to make pallets autonomous an implementing strategy and methodology are required to be employed. Basically, there are several ways to make pallets autonomous. Among them, the pallets can be considered as agents with the capability of simultaneous negotiation and transaction by means of bids and tenders, as a common solution in multi-agent approaches [9] [10]. Nevertheless, this alternative requires some competent negotiation protocols that in complex systems may be difficult to operate. Instead here, the pallets are rather assumed as single entities with no direct negotiation with each other, but they are able to record relevant data and learn from previous behaviors in the system.

This type of pallets contributes to the general concept of autonomous pallets, called learning pallets (Lpallets), in the frame work of our study. This research topic is a contributing input to a universal research over autonomy in logistics at Bremen University, for more information about autonomy in logistics see: www.sfb637.uni-bremen.de. Extensively, the ability of Lpallets can be extended into negotiation level between pallets, the products, and machines, (like multi-agents) in case of requirement. Consequently, the idea of Lpallets enhances the new approach in logistics as autonomous logistic control by means of autonomous logistic objects [11].

Practically, on the merit of Lpallets, intelligent methods are required in order to fulfill the learning and decision making competencies. To this aim, among several possibilities, the radial basis function network (RBFN), as an artificial neural network (ANN) technique, is selected to carry out this task. This type of ANN has some privileges that later follow in details. Finally, for evaluating the performance of Lpallets in real-time scheduling an assembly scenario is modeled by a discrete event simulation structure which employs Lpalltes. The rest of the paper covers the assembly scenario, the applied RBFN technique, and evaluation of the results.

II. LPALLETS

Lpallets as unique objects in logistics have several aspects to be covered which are following:

A. Autonomy for Lpallets

Generally, it is claimed that the prominent specification of autonomous objects is their independency in making their own decisions, in case of alternating circumstances [11]. To realize this, the autonomous objects within a global (an entire) system are arranged in a distributed configuration with rather decentralized authority in a heterarchical structure. In this respect, each object is able to proceed with its local problems, while the threat may be deficits in global awareness of the entire system. In other words, this individuality can be advantageous but sometimes accompanied by some lacks of required information. However, it is claimed here that employment of Lpallets within a closed loop system can partly compensate the missed information in the era of decision making. Since the pallets embedded in assembly systems are constant transport objects, they can collect some data within their rather circulating trips. It means, if the assembly system does not have a strict transient behavior, the individual Lpallets are able to experience the recent performance of the global system by crossing all stations.

This gives the opportunity to the Lpallets to learn the current pattern of the systems' behavior and proceed with that. In spite of the fact that each Lpallets are individual entities, they have their interactions with the system by recording the waiting and processing times in each instance. Assuming the closed system, after a while every individual Lpallet perceives the general attitude of the assembly system as well as other Lpallets. This occurs due to indirect effects of Lpallets' decisions on each other through the system performance. In addition, Lpallets may negotiate with each other, which this case is to be undertaken in further papers.

However, the recognition of patterns by Lpallets can be simply done by use of neural networks, as an intelligent technique for learning the patterns to classify or approximate them later.

B. Artificial Neural Networks

To present the application ANN in supporting the concept of autonomy in logistics, it is enough to consider their learning ability and capability in classifying data as well as approximating functions [12].

ANN span a huge range of networks types which introduction of them is not in the scope of this paper. However, it can be noticed that for studying Lpallets two types of ANN were considered to be examined. First RBFN and second multilayer perceptron networks (MLP). Both networks have some similarities e.g., both are useful for problems in function approximation, data classification, and modeling dynamic systems and time series. Additionally, both networks have iterative training algorithms and both start with initial parameters and get trained by different algorithms e.g., Gauss-Newton, steepest-descent, backpropagation. Furthermore, MLP has defined neurons in its layer while RBFN for each new training pattern requires a new neuron in the hidden layer. However, requirement of several neurons is the weakness of RBFN in comparison with MLP, but RBFN can be trained relatively faster than MLP which is a crucial factor in real-time assembly systems. Finally, in the current paper just RBFN is introduced in details and applied.

C. Radial Basis Function Network

In this work, RBFN is selected to represent the application

of neural networks in Lpallets. RBFN is a two layer neural network with rather Gaussian transfer functions in layer one (hidden) and sigmoid or linear functions in the second layer (output), to aggregate the outputs of the first layer. This type of neural network has a quicker training phase in comparison with other feed-forward networks [13]. Fig. 1 shows the general shape of used RBFN.

$$y_m = \exp\left(\frac{-\left\|x - w_m^{\,1}\right\|^2}{2\sigma_m^{\,2}}\right) \quad (1)$$

Where:

y_m =Output of the m^{th} neuron,

x =The input vector,

$w_m^{\,1}$ = Connecting weight vector of input to m^{th} neuron,

σ_m =Standard deviation in the m^{th} Gaussian function.

$$z_k = \frac{\sum_{m=1}^{M} y_m w_{km}^{\,2}}{M} \quad (2)$$

Where:

z_k = k^{th} Output of the network,

$w_m^{\,2}$ = Connecting weight of the m^{th} neuron to the output k of the network.

While the Lpallets move within the closed loop system they check waiting plus processing times of each station in the system. These times and the station number are the inputs of the neural network. In the training phase of RBFN, for each pallet, every new recognized pattern is distinguished by a new Gaussian function. The new patterns are put to the center of the Gaussian functions, known as kern vectors. Equation (1) defines the output of the hidden layer while (2) shows the output of the output layer. As mentioned, in the hidden layer for each new recognized pattern a new neuron has to be devised. This holds true, whereas in the output layer just three neurons are embedded to classify the inputs into three linguistic terms as good, normal, and bad with their respective membership degrees, inspired by fuzzy system. It is assumed that after some rounds (about ten) each pallet perceives the possible patterns that the system may reflect. In this manner, training of the output layer has a crucial role, since they classify the hidden layer neurons.

In the second layer of RBFN (output layer) the outputs of the first layer (hidden layer) neurons are multiplied by their respective weights and then aggregated together to give the output of the network. After the training phase, when the main possible patterns are recognized, the Lpallets are ready to have a local and, to some extent, global impression of the system. This ability is achieved by training the input weights, the spread of each Gaussian hidden neuron and specifically the outputs weights. The equation (3), (4) defines the applied algorithm, called backpropagation, adjusted to RBFN.

$$w_{km}^{\,2}(t+1) = w_{km}^{\,2}(t) - \alpha_1\left[(-2(a_k - z_k))\frac{y_m}{\sum_{m=1}^{M} y_m}\right] \quad (3)$$

Where:

$w_{km}^{\,2}$ =Output weight of the m^{th} hidden neuron to k^{th} output neuron,

t =Training number

z_k = Real output of the k^{th} output layer,

a_k = Training output of the output layer when is the input,

α_1 = Learning speed.

$$w_m^{\,1}(t+1) = w_m^{\,1}(t) - \alpha_2\left[\sum_{k=1}^{K} -2(a_k - z_k)\frac{w_{km}^{\,2}\sum_{j=1}^{M} y_j - p_k}{\left(\sum_{j=1}^{M} y_j\right)^2} \times \frac{y_m}{\sigma^2 m}\left(x_n - w_m^{\,1}\right)\right] \quad (4)$$

Where:

$p_k = \sum_{m=1}^{M} y_m w_{km}^{\,2}$,$\forall\ k = 1,\ldots,K$,

x_n =The n^{th} input vector,

α_2 = Learning speed.

Here, a quick (half) training system is adopted to adjust just the weights of the network, but not the spread and learning speeds. However, the trainable factors are always exposed to learn new changes. In other words, throughout the running simulation each time a pallet meets a station the respective kern vector adapt itself to the new possible condition. This adaption occurs by substituting the average of the last 3 recorded times of that station to the kern vector. However, since each new recognized pattern built a new RBF neuron, in hidden layer, with embedding the input value as the center of its function, when an input vector is not covered by the range of existing RBF (starting from first neurons to the last one) then this is assumed as a new pattern to the network.

III. Assembly Scenario

In order to reflect the necessity of real-time scheduling and

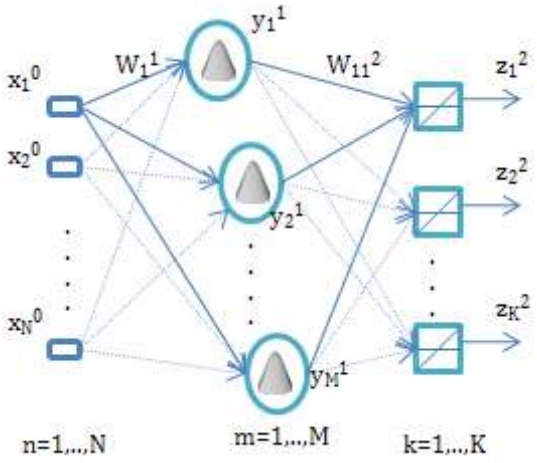

Fig. 1. Topology of the applied RBF neural network.

decision making within a distributed system, an exemplary assembly network is considered to be simulated. However, the assembly network is based on a real platform as a prototype for CRC 637 research cluster at the lab of BIBA GmbH/ University Bremen. Within this simulation model the performance of Lpallets in general allocation (dispatching) decisions is evaluated. In the scenario six working stations are devised that five of them do some operations on products and the last one is the un/loading station of pallets. When the pallets are unloaded from products they wait in a stock for another corresponding products.

There are three types of products each movable by its respective pallets. The semi-finished products come stochastically to the un/loading station and if the pallets are available there they will be released to the system, otherwise wait for available pallets. Number of pallets is constant (12 pallets) throughout the simulation run. This is inspired by Conwip system, which is pre-defined based on the capacity of machines and their buffers [14]. All products must be processed on every station but the sequence of the operations is not fixed. This type of floor operation resembles the open shop problem with fixed number of operations for all products [15]. Except station five and one the sequence of all other stations is changeable. However, either station five or station one is the finalizing operation by considering that if station one is the last operation station five is its predecessor.

However, at the entrance of each station there is a choice of getting in or over taking that. After training RBFNs each pallet as an individual module decides over its own sequence of operations. Fig. 2 represents the modeled assembly system.

Nonetheless, the distributed structure of this problem in terms of machines and pallets besides the stochastic nature of all processes make this allocation problem a case of complex real-time scheduling over time horizon. Every time an individual Lpallet visits a station the waiting time plus processing time together with the station number are recorded to the Lpallat. Then these are the inputs of RBFN to train the system in case of any new pattern recognition. Furthermore, here, the decision making procedure is following.

At the entrance of each station every pallet is a decision maker for its respective operations' sequence. After several round trips of pallets instead of the actual waiting time as input the average of last three records for the corresponding station is taken as the real input to the RBFN. This results in a smoother perception to the dynamic waiting times. However, these inputs through the RBFN are mapped to three categories of outputs, as good, normal, and bad. Each of these terms has its connecting weights (from hidden to output layer) which defines their membership degrees to that judgment. Finally, by summing up the outputs of all stations, in the same Lpallet, the least values of stations, the more priority gets. By doing so, the Lpallet defines its operations' sequence.

It is noticeable that, during the entire simulation each RBFN is trained and by observing any new patter (a value out of the so far covered range) a new neuron is added to the hidden layer. However, for the current problem each pallet may face different patterns, thus, may have alternative number of neurons to the others. Nonetheless, it is seen here that the most getting neurons did not violate the number ten.

IV. Simulation Results

In this section by comparing the performance of Lpallets against a conventional flow shop scheduling, by using first in first out (FIFO) dispatching rule throughout the assembly system, is analyzed. Different scenarios are examined here. Stochastic replenishments (variable intervals) as well as constant intervals in supply of semi-finished products to the entrance (un/load station), besides balanced operation times in every station against unbalanced times, are two variants considered for depicting the performances of such alternating assembly system with the use of Lpallets.

TABLE I
Examined Scenarios with Three Alternatives

Scenario	Process Time of each Station					Supply Inter-arrival Time for each Product Type			Setup Time				
	1	2	3	4	5	1	2	3	1	2	3	4	5
1	Neg. Exp with ß=10 min, for all					Neg. Exp with ß=50 min, for all			5 min, for all				
2	Neg. Exp with ß=10 min, for all					Constant 50 min			5 min, for all				
3	$ß_1$=8, $ß_2$=8, $ß_3$=8, $ß_4$=10, $ß_5$=8					Constant 45 min			5 min, for all				

Furthermore, working time, waiting time, and blocked time of each station as well as average flow time (AFT) of finished products and makespan (completion time) of all orders (150 each type) are some criteria to be compared. Here, the blocked time is the time that a product is asking for operation on a machine but machine is busy. In contrary, the waiting time is the time that machine is waiting for a product to be processed on. Table 1 defines the specification of the three alternative scenarios.

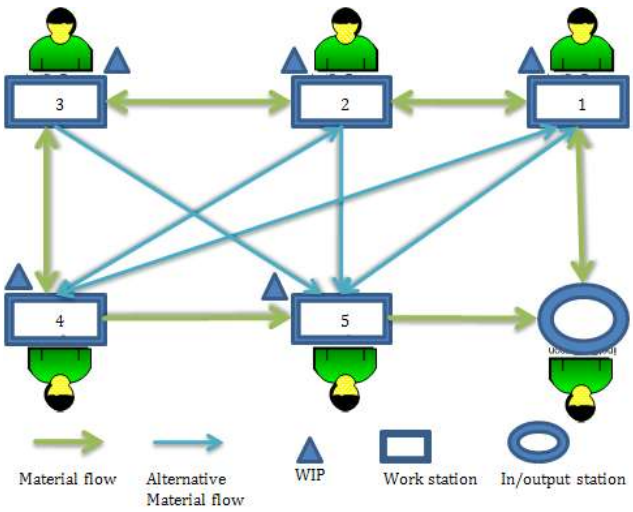

Fig. 2. Description of assembly scenario with 6 stations.

Fig. 3 and Fig. 4 depict the percentage of each time for RBFN and flow shop FIFO respectively, for scenario 1. Fig. 4 and Fig. 5 present the AFT of each product type in RBFN and flow shop FIFO respectively, for scenario 1. Fig. 7 and Fig. 8 show the percentage of each time for RBFN and flow shop FIFO respectively, for scenario 2. Fig. 9 and Fig. 10 reflect the AFT of each product type in RBFN and flow shop FIFO respectively, for scenario 2. Fig. 11 and Fig. 12 present the percentage of each time for RBFN and flow shop FIFO respectively, for scenario 3. Fig. 13 and Fig. 14 depict the AFT of each product type in RBFN and flow shop FIFO, respectively, for scenario 3.

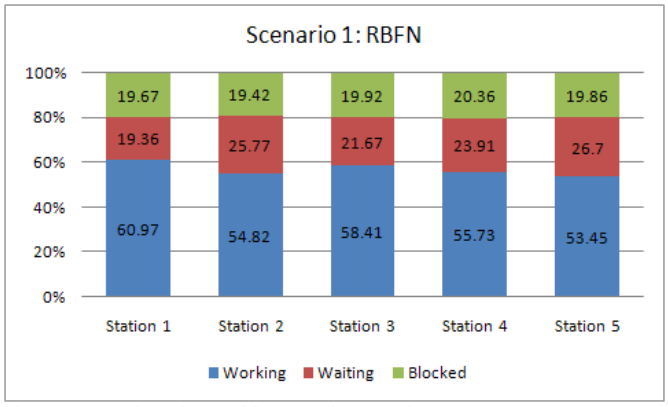

Fig. 3. Processing Times with RBFN in scenario 1.

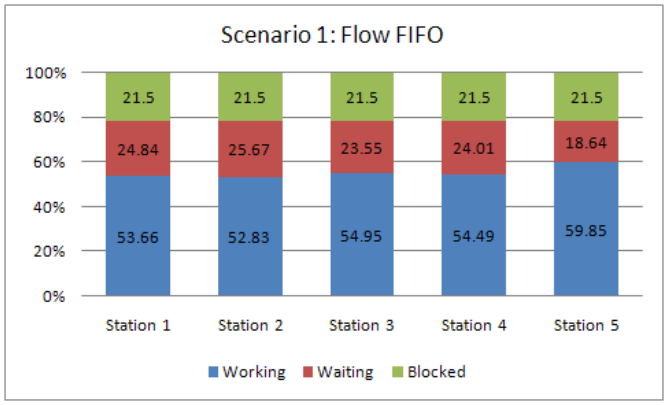

Fig. 4. Processing Times with Flow-Shop FIFO in scenario 1.

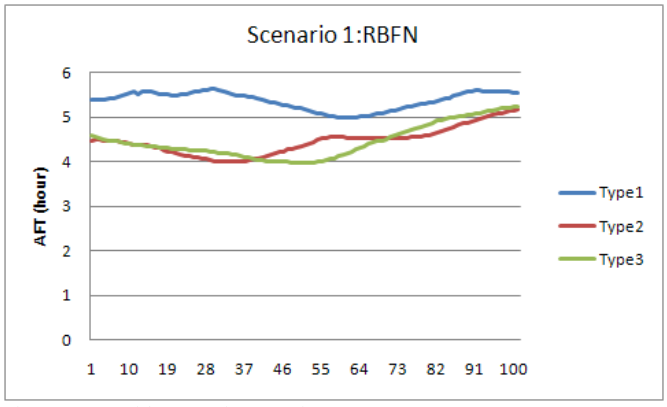

Fig. 5. AFT With RBFN in scenario 1.

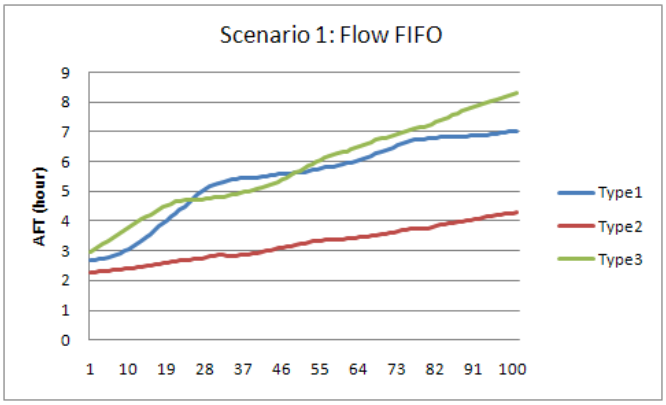

Fig. 6. AFT with Flow-Shop FIFO in scenario 1.

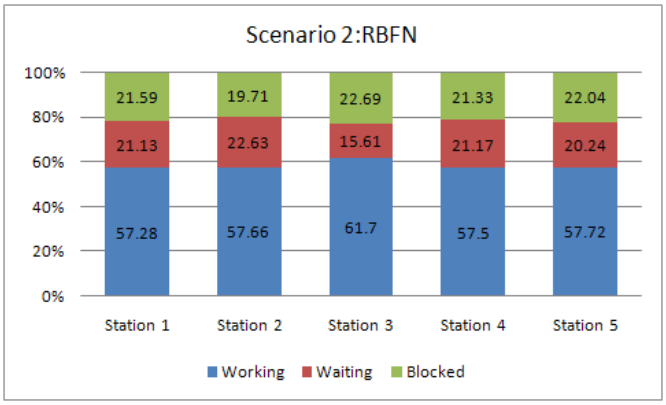

Fig. 7. Processing Times with RBFN in scenario 2.

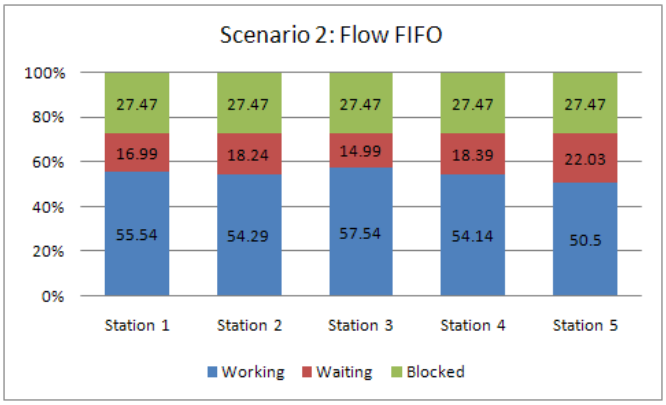

Fig. 8. Processing Times with Flow-Shop FIFO in scenario 2.

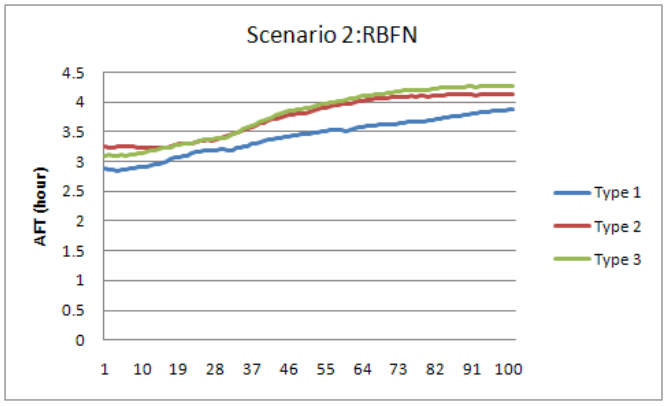

Fig. 9. AFT With RBFN in scenario 2.

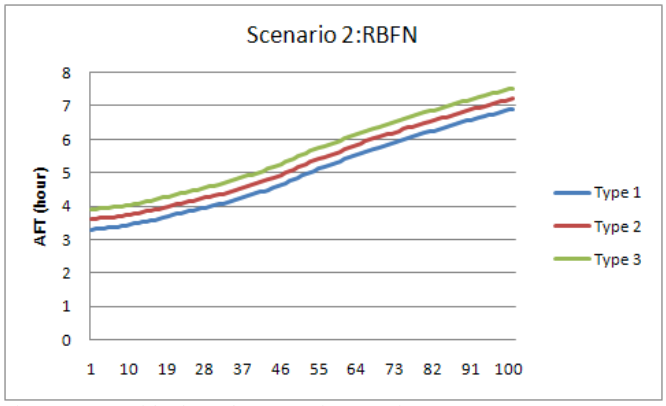

Fig. 10. AFT with Flow-Shop FIFO in scenario 2.

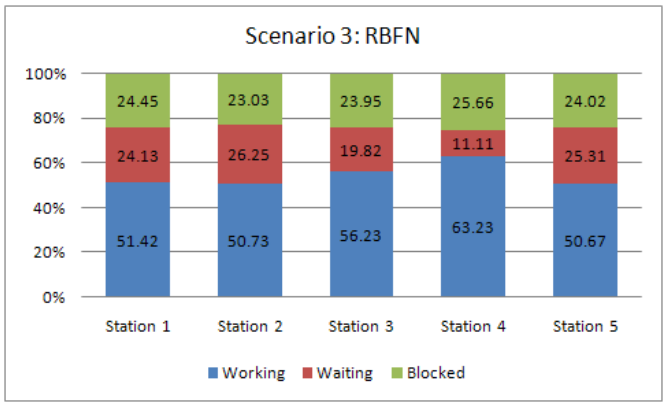

Fig. 11. Processing Times with RBFN in scenario 3.

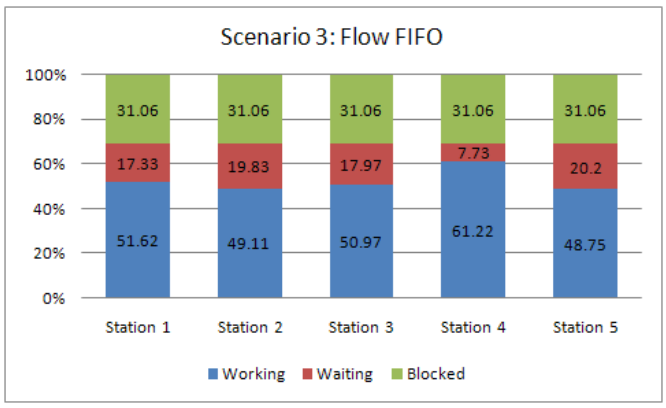

Fig. 12. Processing Times with Flow-Shop FIFO in scenario 3.

As it can be seen in all figures, the performance of RBFN in all scenarios is better in terms of more working time and less blocked or waiting time. In addition, in AFT the use of RBFN shows a convergence in higher experiences which reflects the learning procedure of ANN during the simulation. However, in all FIFO scenarios the AFT exponentially increases.

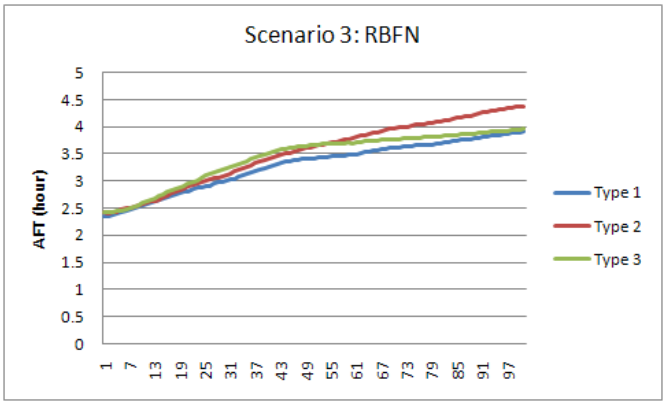

Fig. 13. AFT With RBFN in scenario 3.

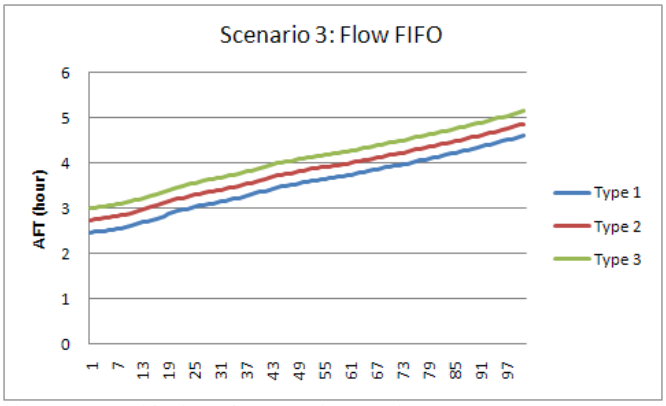

Fig. 14. AFT with Flow-Shop FIFO in scenario 3.

V. Conclusion

In this paper, a new approach to autonomous logistic objects as Lpallets is introduced that uses artificial neural network for learning technique. The specific chosen network is RBFN that recognizes new patterns and adjusts its parameters to new conditions via learning. The Lpallets concept can comply with the requirements of real-time scheduling in practical problems. The concept bears the notion of distributed and decentralized control towards autonomous logistic objects research.

Here, several simulation scenarios were experimented in this paper to show the superior performance of Lpallets against the conventional flow shop FIFO strategy. This was evaluated by some criteria in case of real-time distributed dispatching problem. In all scenarios it was configured that usage of Lpallets result in convergence of AFT as well as higher utilization for stations (more working time and less waiting time).

In this paper a quick training algorithm is undertaken to train the RBFN that trains just the weights. However, in a complete training algorithm the spread of Gaussian functions as well as training rates are tuned as well.

In conclusion, several intelligent learning and decision

making techniques are available to be integrated to Lpallet like fuzzy inference system, genetic algorithm. At the same time by tuning more intelligently the parameters of ANN more accurate performances are more likely.

ACKNOWLEDGMENT

This work is supported by international graduate school (IGS) for dynamics in logistics at Bremen University.

REFERENCES

[1] W. Xiang, and H. Lee, "Ant colony intelligence in multi-agent dynamic manufacturing scheduling," *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 73-85, 2008.

[2] A. Savkin, and J. Somlo, "Optimal distributed real-time scheduling of flexible manufacturing networks modeled as hybrid dynamical systems," *Robotics and Computer-Integrated Manufacturing*, vol. 25, pp. 597-609, 2009.

[3] V. Devadas, F. Li, and H. Aydin, "Competitive analysis of online real-time scheduling algorithms under hard energy constraint," *Real-Time Systems*, Vol.46, pp. 88-120, 2010.

[4] K. Iwamura, and N. Sugimura, "A study on real-time scheduling for autonomous distributed manufacturing systems," *IEEE International Conference on Systems Man and Cybernetics (SMC) Jpn.p.1352-1357*, Oct. 2010.

[5] K. Iwamura, N. Mayumi, Y. Tanimizu, and N. Sugimura, "A Study on Real-time Scheduling for Holonic Manufacturing Systems-Application of Reinforcement Learning," *Service Robotics and Mechatronics*, pp. 201-204, 2010.

[6] R. Maheswaran, , C. M. Rogers, R. Sanchez, and P. Szekely, "Real-Time Multi-Agent Planning and Scheduling in Dynamic Uncertain Domains," *In Proc. 20th Internation Conference on Automated Planning and Scheduling (ICAPS), Can.* p. 16, May. 2010.

[7] C. Wang, H. Ghenniwa, and W. Shen, "Real time distributed shop floor scheduling using an agent-based service-oriented architecture," *International Journal of Production Research,* Vol. 46, pp. 2433-2452, 2008.

[8] M. Tu, L. Jia.Hong, C. Ruey-Shun, C. Kai-Ying, and J. Jung-Sing, "Agent-Based Control Framework for Mass Customization Manufacturing With UHF RFID Technology," *IEEE Systems Journal,* Vol. 3, pp.343-359, Sep. 2009.

[9] Y. Wang, Q. Li, C. Zhao, and H. Xing, "A Multi-Agent System Frame Model for Dynamic Integration," *The 9th International Conference for Young Computer Scientists,* p. 1163-1168, Nov. 2008.

[10] E. Ilie-Zudor, L. Monostori, "Agent-based framework for pre-contractual evaluation of participants in project-delivery supply-chains," *Assembly Automation,* Vol. 29, pp.137–153, 2009.

[11] K. Windt, and M. Hülsmann, "Changing Paradigms in Logistics—Understanding the Shift from Conventional Control to Autonomous Cooperation and Control," in Understanding Autonomous Cooperation and Control in Logistics, Eds. Berlin: Springer, 2007, pp. 1-16.

[12] D.J. Jwo, and C.C. Lai, "Neural network-based GPS GDOP approximation and classification," *GPS Solutions,* Vol. 11, pp. 51-60, 2007.

[13] C. Looney, "Radial basis functional link nets and fuzzy reasoning," *Neurocomputing*, Vol. 48, pp. 489-509, 2002.

[14] N. Li, S. Yao, G. Liu, and C. Zhuang, "Optimization of a multi-Constant Work-in-Process semiconductor assembly and test factory based on performance evaluation," *Computers & industrial engineering*, Vol.59, pp.314-322, 2010.

[15] D. Sha, and C. Hsu, "A new particle swarm optimization for the open shop scheduling problem," *Computers & Operations Research*, Vol. 35, pp. 3243-3261, 2008.